D-039

ビジネスプロセスモデリングにおける時間制約条件の導入

Temporal Constraints in Business Process Modeling

飯塚　隆司 †　　小林　洋 †
Takashi Iitsuka　　Hiromi Kobayashi

１．はじめに

最近，システム開発の分野では業務サービスレベルのコンポーネントを用いた SOA(Service-Oriented Architecture)による開発が注目されている．SOA では業務のプロセスを表す図式として BPMN(Business Process Modeling Notation）が標準的に用いられている．BPMN は BPEL(Business Process Execution Language) や XPDL(XML Process Definition Language)という XML 形式のタグを用いた言語に変換され，アプリケーションからのサービスの呼び出しに利用される．XPDL は BPEL に比べ日本ではあまり普及していないが，BPEL のようにベンダー毎の方言が無く，Web サービス以外の部分も記述できるという特長がある．本研究では，ビジネスプロセスにおいて必要な時間制約条件を Allen の時間関係とカレンダパターンを用いて BPMN に記述し，XPDL に変換後にアプリケーションからのサービス呼び出し時に時間制約条件を渡す試みを行った．

２．ビジネスプロセスの表記

BPMN[1]はビジネスプロセスの表記法で，業務手順を示すワークフローの図式表記法である．特徴としては，ビジネスプロセス固有の複雑性を表現できること，基本要素によるシンプルな記述に段階的にバリエーションを追記して詳細化が図れるようになっていること等が上げられる．

XPDL[2]は，XML ベースのプロセス定義言語であるビジネスプロセスのワークフローを視覚的にも意味的にも保存して，交換するために作られたものである．また，XPDL は図形要素の座標点やメタデータなどの図としての情報を持っている BPMN をサポートしている．図１のように BPMN で表記された図を XPDL に変換することができ，逆変換も可能である．また，BPEL[3]のようにベンダー毎の方言が無く，Web サービス以外の部分も記述できるという特長がある．

３．時間制約条件

Allen の時間関係とカレンダパターンについて説明する．

（１）Allen の時間関係

Allen の７つの時間関係[4]と呼ばれる，二つの時間区間で表されるプロセス間の関係を以下に示す．

①overlaps(x,y) ： start_x < start_y & start_y < end_x
②before(x,y) ： end_x < start_y
③during(x,y) ： start_y < start_x & end_x < end_y
④equals(x,y) ： start_x = start_y & end_x = end_y
⑤meets(x,y) ： end_x = start_y
⑥starts(x,y) ： start_x = start_y & end_x < end_y

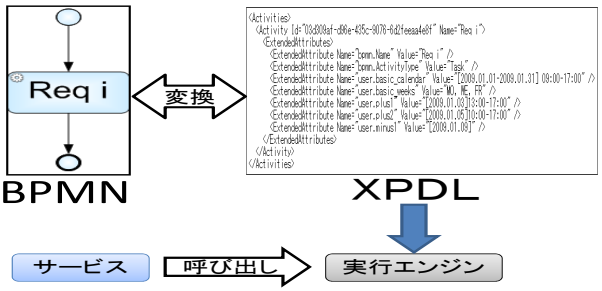

図１　ＢＰＭＮとＸＰＤＬ

⑦finishes(x,y) ： start_y < start_x & end_x = end_y
なお,上記の他，排他を表す下記の関係がしばしば必要になる．
⑧excludes(x,y)：end_x < start_y ‖ end_y < start_x

（２）カレンダパターン

カレンダパターン[5]は，ビジネスがカレンダを基に行われることからビジネスにおける時間関係を表すのに必要で，次のように表すことが出来る．

([T1, T2] W {Weeks} t1 – t2
Plus {{Days} t3 – t4}
Minus {Days})

[T1, T2]の T1 は期間開始，T2 は期間終了の年月日を表す．W {Weeks} t1 – t2 の Weeks はサービス実施の曜日を表し，t1，t2 は各々サービスの開始時間と終了時間を表す．Plus と Minus でサービスの例外日の追加と削除を表し，Days でその年月日，t3 － t4 でサービスの開始時刻と終了時刻を表す．

４．仕様の記述

ビジネスプロセスのワークフローを BPMN で記述し，その中に時間制約条件を次のように記述することにする．

（１）Allen の時間関係は BPMN のレーンの中のユーザ属性内に記述する．

（２）カレンダパターンは BPMN の各アクティビィティのユーザ属性内に記述をする．

以上のようにユーザ属性内に記述した時間制約条件は BPMN から XPDL に変換する際，そのまま保持され，<ExtendedAttribute></ExtendedAttribute>内に記述される．これをアプリケーションのサービス呼び出し時に，アプリケーション側で取り込み，処理を行うようにする．なお，今回 BPMN から XPDL への変換には ITP Process Modeler for Microsoft Visio を使用した．

５．適用例

複数のソフトウェア開発プロジェクトを対象にした簡単な適用例を次に示す．図１に BPMN での記述を示す．

† 東海大学 工学研究科， Tokai University

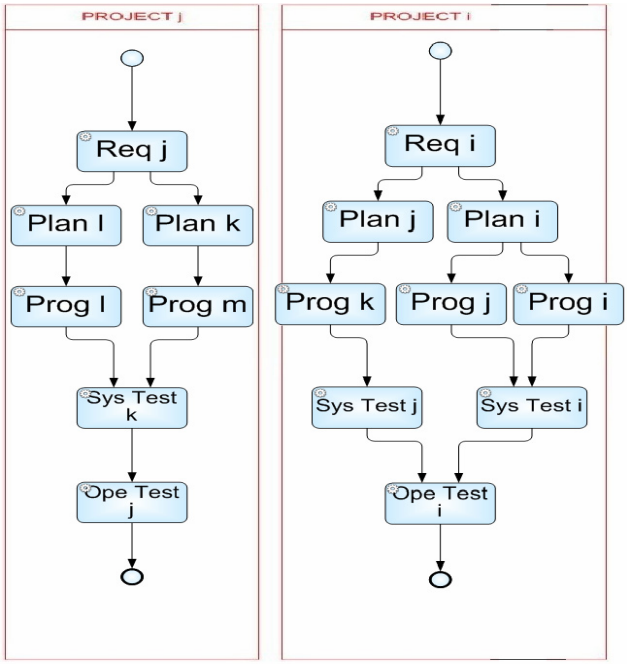

図２　ソフトウェア開発のプロジェクト管理

図２は，アクティビティと呼ばれる作業単位の流れの関係を表しており，この図に Allen の時間関係とカレンダパターンを記述する．

（１）Allen の時間関係の適用例

次のようなケースを考える．

ａ．プロジェクト ｉ と ｊ の要求分析を表すアクティビティ Req_i と Req_j は，人員の制約から同時に行うことが出来ないものとする．

excludes(Req_i , Req_j)

ｂ．システムテスト Sys_Test_i は Sys_Test_j より先に始め，先に終わらせなければならないものとする．

overlaps(Sys_Test i, Sys_Test j)

ｃ．プログラミング Prog_i , Prog_j および Prog_k は同時に始め，同時に終わらせたい．（３項関係に拡張）

equals(Prog_i , Prog_j, Prog_k)

（２）カレンダパターンの適用例

次に，カレンダパターンの実装例を示す．図２の Req_i に３．（２）のカレンダパターンを基に BPMN のユーザ属性内に次のように記述を行う．

ユーザー定義の属性	
basic_calendar	[2009.01.01-2009.01.31] 09:00-17:00
basic_weeks	MO, WE, FR
plus1	[2009.01.03]13:00-17:00
plus2	[2009.01.05]10:00-17:00
minus1	[2009.01.09]

これを ITP Process Modeler for Microsoft Visio により XPDL に変換すると図３のようになる．アクティビティは <Activities></Activities>内に記述され，その中にカレンダパターンが<ExtendedAttribute></ExtendedAttribute>内に記述される．Name の部分にはカレンダパターンの時間制約条件内の名前が，Values の部分には時間制約条件の値が記入される．

```
<Activities>
  <Activity Id="03d309af-d86e-435c-9076-6d2feeaa4e8f" Name="Req i">
    <ExtendedAttributes>
      <ExtendedAttribute Name="bpmn.Name" Value="Req i" />
      <ExtendedAttribute Name="bpmn.ActivityType" Value="Task" />
      <ExtendedAttribute Name="user.basic_calendar" Value="[2009.01.01-2009.01.31] 09:00-17:00" />
      <ExtendedAttribute Name="user.basic_weeks" Value="MO, WE, FR" />
      <ExtendedAttribute Name="user.plus1" Value="[2009.01.03]13:00-17:00" />
      <ExtendedAttribute Name="user.plus2" Value="[2009.01.05]10:00-17:00" />
      <ExtendedAttribute Name="user.minus1" Value="[2009.01.09]" />
    </ExtendedAttributes>
  </Activity>
</Activities>
```

図３　カレンダパターンのＸＰＤＬへの変換例

６．使用例

時間制約条件を用いることにより，例えば次のようなプロジェクトスケジュール管理に応用することが考えられる．

（１）スケジュールチェック

プロジェクトの計画策定時や実行時のスケジュールチェックを行う．各プロジェクト間の整合性を Allen の時間条件を使用しチェックしたり，カレンダパターンから管理者やメンタ（特殊技能を持つ人）が勤務中で助言等のサービスを受けることができるか問い合わせる．

（２）スケジュールモニタリング

プロジェクトの実行時にスケジュールを監視する．各アクティビティの終了直前に，プロジェクトのリポジトリへの成果物の送付の注意を促したり，終了時期を過ぎた際の督促通知をプロジェクト参加者に送る．

７．おわりに

BPMN に Allen の時間関係とカレンダパターンに基づく時間制約条件を記述するための方式を示した．BPMN は XPDL に既存のツールにより自動変換することが可能である．この際，XML 形式で書かれた時間制約条件をアプリケーションのサービス呼び出し時にアプリケーションに渡すことにより，アプリケーション側ではこれを使用したサービスを作成することが可能となる．例えば，BPMN でのビジネスのスケジュール作成時のスケジュールチェッカや，ビジネスプロセス実行時のモニタとして用いることが考えられる．これらの実装については，現在計画中である．

参考文献

[1]BPMN Home page: http://www.jsys-products.com/iwaken/bpmn/pub/BPMN.pdf
[2]XPDL Home page: http://www.wfmc.org/xpdl.html
[3]BPEL Home page: http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf
[4]J.Allen, Maintaining knowledge about temporal intervals, CACM, Vol.26, No.11, pp.832-843, 1983.
[5]C. Bettini, S. Jajodia, and S.X. Wang , Time Granularities, Springer, 2000.